Neural Network Fusion of Color, Depth and Location for Object Instance Recognition on a Mobile Robot

Louis-Charles Caron, David Filliat, Alexander Gepperth

Neural Network Fusion of Color, Depth and Location for Object Instance Recognition on a Mobile Robot

Louis-Charles Caron and David Filliat and Alexander Gepperth

Robotics and Computer Vision team, École Nationale Supérieure des Techniques Avancées, 828, Boulevard des Maréchaux, 91762 Palaiseau Cedex, France

Abstract. The development of mobile robots for domestic assistance requires solving problems integrating ideas from different fields of research like computer vision, robotic manipulation, localization and mapping. Semantic mapping, that is, the enrichment a map with high-level information like room and object identities, is an example of such a complex robotic task. Solving this task requires taking into account hard software and hardware constraints brought by the context of autonomous mobile robots, where short processing times and low energy consumption are mandatory. We present a light-weight scene segmentation and object instance recognition algorithm using an RGB-D camera and demonstrate it in a semantic mapping experiment. Our method uses a feed-forward neural network to fuse texture, color and depth information. Running at 3 Hz on a single laptop computer, our algorithm achieves a recognition rate of 97% in a controlled environment, and 87% in the adversarial conditions of a real robotic task. Our results demonstrate that state of the art recognition rates on a database does not guarantee performance in a real world experiment. We also show the benefit in these conditions of fusing several recognition decisions and data from different sources. The database we compiled for the purpose of this study is publicly available.

Keywords: Semantic mapping, indoor scene understanding, instance recognition, mobile robotics, RGB-D camera

1 Introduction

For robots to accomplish useful tasks, they must have the capacity to understand their environment. Endowing robots with the ability to recognize previously seen objects is one step to take them out of the labs into the real world. Our research focuses on assistive robotics, where an autonomous robot shares the home of its owner and helps him with his daily chores. In this context, it is important for the robot to recognize each piece of furniture and commonplace objects in their home. This is called object instance recognition, as opposed to object category recognition which aims at identifying an unknown object's class. The context of autonomous mobile robots brings limits on processing power and energy. The lighter the algorithms, the more reactive the robot can be, and the longer it

can operate without having to recharge. The algorithms described in this paper perform object instance recognition and were specifically designed to be light-weight.

Our recognition algorithm relies on data provided by an RGB-D camera (cameras providing color and depth information). These cameras have become omnipresent in robotics labs because they are affordable and give valuable information about a robot's surroundings. They are however noisy, giving imprecise distance measures, and some models can suffer from data synchronization issues, see figure 4. In our experiments, when the camera is moved quickly as happens when it is mounted on a mobile robot, a shifting occurs between the depth and color image. Images also tend to be plagued with motion blur. Common RGB-D cameras also perform very badly on reflective, transparent and dark surfaces. In real-life robotics experiments, data is further affected by partial views, occlusions, viewpoint variance and illumination changes. However, real robotic experiments allow for further processing to be done on top of single image recognition. For instance, performances can be improved by cumulating recognition scores when an object is seen several times.

The contribution of this paper is threefold. We describe an integrated RGB-D scene segmentation and object instance recognition algorithm for mobile robots. We provide the database we compiled to train the algorithm consisting of about 31200 RGB-D images of 52 common objects (600 per object). We propose a benchmark robotic experiment to evaluate the recognition of objects when they are seen in a different context as when they were learned. Our recognition method copes with all aforementioned problems and is light-weight enough to be run on an autonomous robot while it performs other tasks. In our experiment, a mobile robot performs a semantic mapping task in which it must map its environment and annotate the map with information about the objects it encountered. This experiment involves many challenges which are not encountered when using offline database for performance evaluation. As by the focus of our research, we concentrate on rather large objects lying on the ground because they can serve for navigation and indoor scene understanding. A semantic map obtained with our algorithms is shown in figure 1. This paper extends the work of [11], adding improved recognition capabilities, reliable scene segmentation and thorough analysis of performance. For the purposes of this article, our benchmarking efforts will focus on recognition. A particular interest of our investigations has been the evaluation of the fusion of color and depth information for improving recognition accuracy. The fusion is done with a feed-forward neural network.

This paper is structured as follows. The state of the art in the domains of point cloud segmentation and object recognition is presented in section 2. The database compiled for the purposes of our experiments and information about the physical implementation of our methods are detailed in section 3. Section 4 describes our segmentation and recognition algorithms. Our results are shown in section 5 and they are commented in section 6.

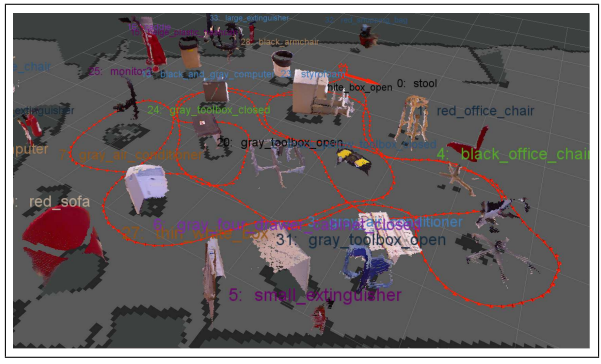

Fig. 1. The semantic map resulting from the online experiment.

2 Related Work

Traditionally, the steps of segmentation and recognition are deemed to be essential, and this is indeed the approach we use in this article. However, segmentation-free recognition approaches have been demonstrated to be feasible and computationally efficient [31, 14], though not directly in the field of robotics where recognition problems typically exhibit a very high number of object instances. Here, we will briefly review some related work for both segmentation and recognition while being aware that a field as vast as this one can only be touched lightly within the scope of this article.

2.1 Segmentation

Point cloud segmentation is a field of ongoing research, ignited by the recent advances in 3D sensing technology. The methods related here are roughly introduced in the order of the strength of the hypotheses they make on object shapes, from model-based methods to model-free ones.

Model-based techniques find prototypical shapes in an image and fit geometric models to estimate their exact pose. Such an approach is presented in [24]. Working even with partial views, they find the shape of objects by fitting geometric models like cylinders and planes. Missing data can be filled once the right shape model is found and fitted. After using a surface reconstruction technique, the final model of an object is a hybrid shape and surface description.

Depending on weaker shape hypotheses, [30] segments highly cluttered scenes by analysing point normals. First, raw depth images are spatially and temporally filtered and the point normals are computed. The scene is over-segmented into smoothly curved surfaces by thresholding the normals orientation difference of neighboring points. The segments are joined based on geometric considerations. The method is model-free, but biased to work when the scenes consist of simple box- or cylinder-shaped objects. An approach using multi-modal data is presented in [7], combining image and range data to form a hierarchy of segments

and sub-segments. These segments are rated according to various "structureness" measures in order to retain the best object candidates. Using a similar rating concept, [15] detect multiple objects in Kinect Fusion maps of cluttered scenes. First, the scenes are over-segmented using a graph-based algorithm by [10]. Weights in the graph are computed from the dot-product of the normals of two points and the fact the they are part of a concave or convex surface. The segments are then rated for "objectness" based on different measures such as compactness and symmetry. These measures can further be used for object identification. Dubois et al. [8] propose an energy-based semantic segmentation method and compare it to a geometric method. Their method uses a Markov Random Field and relies on weak hypotheses of smoothness over appearance and labels. It is more generic than the geometric approach but its precision-recall figures are not as good as those of the carefully tuned geometric method when used specifically in indoor scenarios.

Finman and Whelan [12] compute the difference between two Kintinuous maps of a given scene where in one of the maps, an object was either added or removed. Taking into account the angle of view when the scenes were shot, they can obtain the 3-D mesh of the object. Then, they train a segmentation algorithm to obtain the parameters to extract this object from the particular scene where it was seen. Once learned, the object-specific parameters can serve for object detection. This method requires to store a detailed representation of previously seen scenes to accomplish the differentiation.

2.2 3D Object Recognition

The recognition of objects from three-dimensional data is well studied in the literature, see [6, 18] for survey. Many methods expect a segmented object candidate which should be matched against templates in a database of previously registered objects. At the most fundamental level, proposed methods can be grouped into holistic and local approaches. Of the former, a prominent example is iterative closest point estimation (ICP) [34], which can match object candidates to templates if a rough alignment between the perceived object and at least one template exists. The Generalized Hough Transform [9] can be a useful tool, especially for simple objects like cylinders or spheres, see, e.g., [22]. Both these techniques restrict recognition to specific object types, known in advance.

If the object class is unknown, more general methods for the holistic description of objects need to be used: an example is [32] where histograms of normal orientations between randomly chosen point pairs in the object candidate are computed, resulting in a holistic descriptor of object shape. Another notable holistic approach [20] attempts to find constant object signatures in views of object candidates that were taken from different directions. Histograms of pairs of points and normals describe very well the shape of objects and are used in the present work as one of the features fed to our learning algorithm.

Mueller et al. [19] use rules on segments size, position and alignment to merge segments into parts and parts into objects. They demonstrate good recognition results in a cluttered and disorganized scene, but with only 3 object classes.

Bo et al. [3] use unsupervised hierarchical matching pursuit to learn features suitable for object recognition. The method seems very powerful and gives excellent results for generalization over classes and instance recognition. The instance recognition evaluation is realized with objects seen from a different angle of view, but with no occlusions or change in the lighting conditions.

2.3 RGB-D Databases

Databases of RGB-D images of objects already exist. The RGB-D Object Dataset [16] and 2D3D dataset [5] are good examples. The RGB-D dataset is very similar to our own. It contains images of a large number of objects shot from different viewing angles and under controlled conditions. It additionally contains videos of scenes where a certain number of these objects can be found, under different lighting conditions and in challenging situations. The conditions in which the videos were taken however do not allow for our segmentation algorithm to be used. Also, it contains mainly small objects like bowls, bottles or cereal boxes whereas we focus on furniture like chairs and trash cans. Most of the existing methods which were tested on these database only provide performance for recognition of objects shot in controlled conditions and dismiss the video data.

2.4 Summary

For the purpose of this study, elaborate segmentation techniques are not beneficial. The most important points here being to work directly on the image provided by the RGB-D camera, and to preserve processing power. This eliminates methods working on Kinect Fusion or Kintinuous maps [12] and ones implying temporal filtering [30]. Many other techniques are simply too slow, need to be run on high-end power hungry computers or GPUs to run at a decent speed [3, 2, 16], or do not provide speed measures to compare with our method. Model-based segmentation techniques are too constraining for our setup. The methods measuring how much segments look like objects [15] does not aim at the detection of the kind objects that we use. Such techniques essentially prefer small, compact objects, over-segmenting human size complex objects like office chairs. It seems challenging to find measures that would work well for all everyday objects. Most techniques relying on smoothness and slowly changing curvature assumptions often are only demonstrated on very typical boxes and cylinders [17, 30], which does not fit the context of our experiments. No one method focuses both on segmenting and recognizing objects in different contexts with algorithms simple enough to smoothly run on a mobile robot.

Our segmentation algorithm is very close to that of [1]. Whereas they operate in a table-top setting and rely on RANSAC to locate a plane supporting the objects in a scene, our RGB-D camera is at a known position with regard to the floor plane which can thus be identified geometrically. Once the main plane is removed, objects are found by using Euclidean clustering on the remaining points. As noted by the authors, some objects tend to be over-segmented by this procedure. To address thus issue, we project the points on the floor plane before

proceeding with the clustering. [1] also follow the segmentation with an object recognition phase, but they focus on the improvement provided by the knowledge of the co-occurrence statistics of objects as learned from public databases.

3 Methodology

3.1 Database

We compiled a database of RGB-D images (and data from other robot sensors) of 52 objects, shot from 6 viewing angles. The data was collected by a robot as it autonomously moved back and forth in front of the objects. For each angle of view, 100 snapshots were taken from a distance varying between 1 and 4 meters. The objects lay on the floor, in open space. Because the robot moves during data acquisition, shifts sometimes occur between the color and the depth image and some shots are partial views of an object. As these conditions reflect the situation in which the algorithms will be tested, imperfect data were kept in the database. Only shots were the object does not appear were manually removed. The RGB-D images were processed by our segmentation algorithm, described in section 4.1, to produce appropriate data for the object recognition algorithm. This data is referred to as offline data. It was acquired with the room lights on and the windows blinds closed, during summer time. An example image of 19 objects of the database are shown in figure 2.

Fig. 2. Examples of cropped images of 19 objects from the offline database. Segmentation errors and sensor imperfections can be seen on the left hand side computer and the red office chair.

3.2 Robotic Experiment

A second, much smaller, database was also collected for testing purposes. We chose 22 objects from the offline database and laid them on the floor, apart

from each other. The same robot that was used for building the database was, this time, manually controlled to wander around among the objects at the same time as it executed diverse tasks. These tasks are simultaneous localization and mapping, scene segmentation of the RGB-D image, object recognition of the segments, and display of the resulting semantic map (a map containing the objects found and their label). This experiment was conducted in the same room where the offline database was collected, but in a different part of the room, see figure 3. The lights were on, the window blinds open, and it was winter. This database contains a total of 135 segmented objects, with at least one occurrence of an object from 18 object instances from the offline database (due to some segmentation errors, some of the 22 selected objects are never seen). Figure 4 shows some examples of the difficulties encountered in this experiment.

Fig. 3. The room in which the online experiment is conducted, with some of the objects from the offline database.

3.3 Implementation Details

We use a pioneer 3DX robot, with a Kinect RGB-D camera mounted at 1 meter from the ground and tilted slightly downward. The robot is equipped with an Hokuyo laser range finder. All software is run on a single Toshiba Tecra laptop computer (Intel Core i5, 3GB RAM) with Ubuntu 12.04. We use ROS Hydro Medusa [21] for integration, the Point Cloud Library 1.7 [25] for handling RGB-D images, OpenCV 2.4 [4] for computing color and SIFT features and PyBrain 0.3 [28] for the feed-forward neural network.

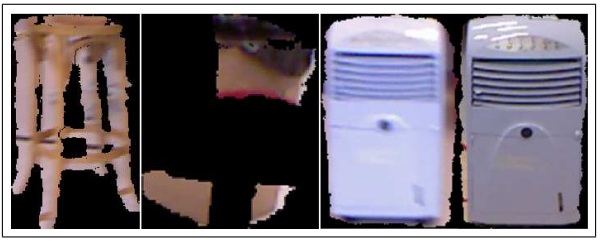

Fig. 4. Examples of difficult online object recognitions. The stool on the left hand side is blurry, and the binary mask generated from the depth segmentation is not aligned with its RGB image. The trash can in the center is occluded by an office chair. The left hand side gray air conditioner (from the online experiment) has a different color than that on the right hand side (from the database). All three objects are correctly recognized by our algorithm.

4 Algorithms

The algorithms described here improve from what is related in [11], providing more stable segmentation and reliable object recognition.

4.1 Scene Segmentation

For scene segmentation, we only use the depth information from the RGB-D camera. The image is converted to a PCL point cloud, points lying farther than 3 meters from the RGB-D camera are filtered out and point normals are computed. The segmentation procedure is split in 4 steps: floor plane removal, wall removal, clustering and filtering. As an offline calibration procedure, the position of the RGB-D camera with respect to the floor is estimated. For this purpose, we previously ran random sample consensus (RANSAC) [13] with a plane model while placing the robot in such a way that the floor covers at least half of the RGB-D image.

Since the camera is not perfectly stable when the robot moves, the floor plane position estimate must be refined for every acquired point cloud. Points lying either 20 cm above or below the estimated floor plane and having a normal perpendicular to this plane (dot-product of the point's and the floor plane's normal higher than 0.98) are identified. From these points, a mean square estimate of the current floor plane's coefficients is computed. All points lying higher than 5 cm over this plane are passed to the next processing steps.

In the wall removal step, points lying on the walls are removed. To find these points, a RANSAC is used again with the added constraint that the plane model must be perpendicular to the current floor plane. All point lying less the 5 cm away from a plane found by the RANSAC, are removed from the point cloud and the process is repeated until no planes are found. The size of each found plane is computed and if it is large enough, it is considered as a wall, otherwise it is reintegrated to the point cloud. The size threshold used in our experiments is 60 000 points.

The next step is to cluster the remaining points into groups that will be considered as objects. The clustering is done based on euclidean distance, grouping any point closer than 10 cm from each other. The clustering is a costly operation, because it has to identify each point's neighbors. The point cloud is passed through a voxel grid filter with 1 cm resolution beforehand to speed up the process. Additionally, the point cloud is projected to the ground floor. As explained in section 4.1, this is done to ensure that complex-shaped objects do not get under-segmented. The clustering is performed and groups containing more than 100 points are kept.

The last step is a filtering operation that removes groups of points that touch a border of the RGB-D image. To maintain the highest possible accuracy, each remaining cluster is "de-voxelized" (reconstructed from the original point cloud). The segmentation algorithm's outputs are the point cloud, rectangle-crop image and a pixel accurate binary mask of each object. As an example, figure 5 shows the segmentation of a part of the scene shown in figure 3.

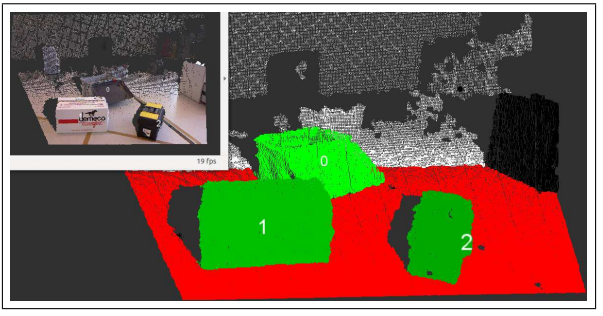

Fig. 5. Segmentation of a scene. The floor is red and the segmented objects are green. The black object on the right hand side is dismissed because it touches the border of the image.

4.2 Object Recognition

Features The object recognition algorithm relies of features computed both on the object's point clouds and images. We use 3 different features: vocabulary of SIFT features, transformed RGB histograms [27] and point feature histograms [26].

A vocabulary of 100 SIFT features is computed by L2-clustering of the SIFT computed from the whole offline database. SIFT features computed from an object's image are matched to the vocabulary and a 100-bin histogram of word occurrence is built.

The transformed RGB histograms are normalized histograms computed on the entire masked image of an object. There are 16 bins for each RGB channel

and each channel is separately normalized to zero mean and unit variance. This yields a 48-bin histogram.

The point feature histogram is computed by using 10 000 randomly selected pair of point from an object's point cloud. A point feature is computed for each pair with the distance measure normalized to the size of the object (the largest distance separating two points from the object's point cloud). The angular features and the distance are discretized to 5 levels and a 625-bin histogram is compiled.

The histograms can be used independently or concatenated to test the influence of each on performance.

Learning and Decision Making The features computed on the training dataset serve as training data for a 3-layer feed-forward neural network. The size of the input layer depends on the features used, the hidden layer has 50 neurons and the output layer has 52, the number of objects in the database. The hidden layer has a sigmoid activation function and the output layer has a softmax activation. Neural network training is done using Rprop [23] training algorithm with early stopping, all layers are fully connected and with a bias unit. The neural network's output is a score giving the confidence for the unknown object to be either one of the 52 training instances. For offline and simple online tests, objects are given the label of the output neuron with highest score.

Map-Aware Recognition The semantic mapping experiment brings many challenges, but has the advantage of allowing the robot to locate objects in space. In the map-aware online tests, an object's location and score is stored for accumulation. If another object is found within 30 cm of an already stored object, the recognition decision is based on the sum of their scores weighted by their size (the number of points in their point cloud).

4.3 Performance

The segmentation and object recognition algorithm runs at a rate of 3 Hz using the hardware and software setup described in section 3.3.

5 Results

Results are presented for different training and testing schemes. Four experiments were conducted: simple offline, one-angle-out offline, simple online and map-aware online recognition rates. All results are shown in table 1, comparing recognition rates obtained by using different combinations of features. All rates are computed by ignoring misclassifications induced by errors in the segmentation step.

The simple offline measure refers to the recognition rates obtained from training the neural network on 90% of the images from the database and testing on

the remaining, using data from all 6 angles of view. In the one-angle-out offline experiment, the network was trained with angles of view 0°, 60°, 120°, 180° and 240° and tested on the 300° angle of view.

The online recognition rate relates to the performance of the recognition algorithm trained on the whole offline data and tested on the online data. As the online data was taken in a different situation than for the offline data, see figure 4, online recognition rates are much lower than offline recognition rates. For these tests, no data from the online database was used for training the learning algorithm. Of course, online performance measures dictated many of our high-level design choices, especially the choice of the color features. In the simple online test, each object encountered by the robot is identified individually. In the map-aware online test, the object's scores are cumulated before a decision is taken, as explained in section 4.2. Figure 6 shows the confusion of the map-aware online test when using all three features.

Table 1. Recognition rates for the simple offline, one-angle-out offline, simple online and map-aware online experiments using different combinations of features.

Features			Simple	One-angle-	Simple	Map-aware
SIFT	Color	Depth	offline	out offline	online	online
	✓		92%	72%	21%	11%
✓			79%	54%	33%	64%
✓	✓		92%	75%	39%	63%
		✓	94%	85%	**70%**	81%
✓		✓	96%	84%	62%	79%
	✓	✓	96%	**89%**	69%	**87%**
✓	✓	✓	**97%**	**89%**	**70%**	**87%**

6 Discussion

6.1 Scene segmentation

The scene segmentation algorithm relies on the stability of the physical configuration of the robot. In an indoor laboratory or apartment setting, this assumption will hold most of the time, but it will fail in certain situations. Staircases, for example, cannot be handled with our method. Otherwise, parts of the RGB-D images belonging to the floor plane are accurately identified and removed. This is true even if the robot accelerates brusquely, bumps into obstacles and oscillates, as happens during operation.

The wall detection step of the segmentation is more problematic. If we only use RANSAC to detect these planes, the results are not reliable because aligned objects can form planes that will be labelled as walls and removed. For this

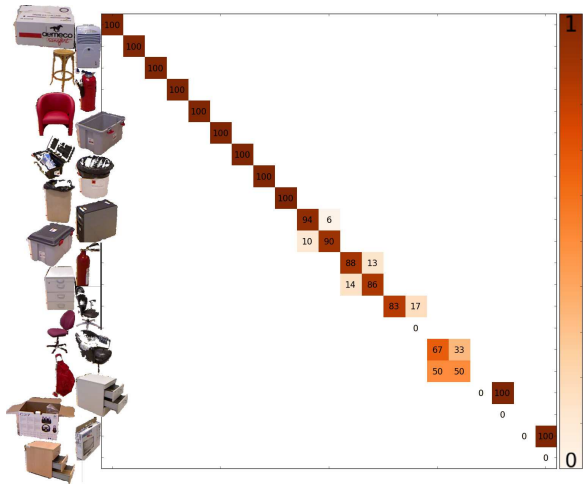

Fig. 6. Confusion matrix for the map-aware online robotic experiment using all three features. Please note that some objects present in the room do not appear in the confusion matrix because they were badly segmented.

reason, we use the RANSAC only to remove very big planes (with more than 60000 points), which really only appear when the robot faces a close-by wall. We had to rely on a more drastic measure to handle the smaller parts of walls: removing all points belonging to an object that touches a border of the RBG-D image (step 4 of the segmentation). This makes sure that walls and objects too big to entirely fit in a single image are not considered for object recognition. It has the drawback of also removing real objects that lie close to a wall. The RANSAC detection of wall, in a perfect setting, allowed us to detect these objects. We plan to address this issue be using information from the map to detect walls as a replacement for the RANSAC.

Our segmentation method does not produce many wrong candidates for object recognition, but rather tends to under-segment objects. Objects lying too close together will systematically be merged. This is the main problem of our method and we hope to use object recognition to solve it, as explained in the next section 6.3.

6.2 Object Recognition

The results from our object recognition demonstrate that we compare to the state of the art, with the added benefit of not being resource hungry. The one-angle-out offline experiment was conducted specifically to ease the comparison with other techniques such as [3], where recognitions are done on shots taken from a previously unseen angle. Of course, as the data and experiments are not exactly the same, it is difficult to draw conclusions from the numbers.

Our experiments also demonstrate that such offline measures are not a reliable indicator of the performance of a system during a real robotic experiment. During our experiments, we went through the process of testing different color features, inspired by the work of [27]. Even though the offline and online data we used in this paper do not differ much, the change was sufficient to make several of our attempts fail. In the end, the least discriminative color feature, also the one yielding the lowest performance on the offline experiments, gave the best results during the online tests. Still, the map-aware results show that these less discriminative features help in a real-life setting.

This is the last point we wanted to bring forward, that the fusion of texture, color, shape and position information is beneficial for recognition, especially in adversarial conditions. Table 1 shows that in our setting, it is the position fusion, done in the map-aware experiments, that is the most beneficial form of data fusion. Fusion of other modalities does not seem to ensure better results. We believe this is due to our database, in which almost every object can be distinguished based on shape only. Few objects, like the two sofas are identical but only differ in color. And in their case, we observed that the different colors affected the Kinect's depth sensor in a way that can be captured by the shape descriptor and allowed correct recognition.

6.3 Future Work

In this paper, the segmentation step is purely geometric, and cannot separate two objects if they touch each other. In the future, we are interested in exploring the use of recognition results to improve the precision of the scene segmentation. We will develop a hybrid bottom-up (geometric segmentation) and top-down (segmentation of recognized objects) to generate candidate objects and refine these segments further. As our segmentation algorithm has a tendency to provide under segmented objects and not many false candidates, we can hope to improve results in this way. Techniques based on the analysis of locally co-occurring visual words [29] or the Hough transform [33] seem promising.

Acknowledgment

The authors thank Jérôme Béchu for his help in the development of the visualization software. Louis-Charles Caron is funded by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Ali, H., Shafait, F., Giannakidou, E., Vakali, A., Figueroa, N., Varvadoukas, T., Mavridis, N.: Contextual object category recognition for RGB-D scene labeling. Robotics and Autonomous Systems 62(2), 241–256 (Feb 2014)
2. Anand, A., Koppula, H.S., Joachims, T., Saxena, A.: Contextually Guided Semantic Labeling and Search for Three-Dimensional Point Clouds. The International Journal of Robotics Research 32(1), 19–34 (2013)

3. Bo, L., Ren, X., Fox, D.: Unsupervised feature learning for RGB-D based object recognition. Experimental Robotics pp. 1–15 (2013)
4. Bradski, G., Kaehler, A.: Learning OpenCV: Computer vision with the OpenCV library. O'reilly (2008)
5. Browatzki, B., Fischer, J., Graf, B., Bulthoff, H.H., Wallraven, C.: Going into depth: Evaluating 2D and 3D cues for object classification on a new, large-scale object dataset. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). pp. 1189–1195. IEEE (Nov 2011)
6. Campbell, R.J., Flynn, P.J.: A Survey Of Free-Form Object Representation and Recognition Techniques. Computer Vision and Image Understanding 81(2), 166–210 (Feb 2001)
7. Collet, A., Srinivasa, S., Hebert, M.: Structure discovery in multi-modal data: a region-based approach. In: IEEE International Conference on Robotics and Automation (ICRA) (2011)
8. Dubois, M., Rozo, P.K., Gepperth, A., Gonzalez, F.A., Filliat, D.: A comparison of geometric and energy-based point cloud semantic segmentation methods. In: Proc. of the 6th European Conference on Mobile Robotics (ECMR) (2013)
9. Duda, R., Hart, P.: Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM 15(April 1971), 11–15 (1972)
10. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision 59(2), 167–181 (Sep 2004)
11. Filliat, D., Battesti, E., Bazeille, S., Duceux, G., Gepperth, A., Harrath, L., Jebari, I., Pereira, R., Tapus, A., Meyer, C., Ieng, S.H., Benosman, R., Cizeron, E., Mamanna, J.C., Pothier, B.: Rgbd object recognition and visual texture classification for indoor semantic mapping. In: 4th Annual IEEE conference on Technologies for Practical Robot Aplications (2011)
12. Finman, R., Whelan, T., Kaess, M., Leonard, J.J.: Toward lifelong object segmentation from change detection in dense rgb-d maps. In: Mobile Robots (ECMR), 2013 European Conference on. pp. 178–185. IEEE (2013)
13. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
14. Gepperth, A.: Object detection and feature base learning by sparse convolutional neural networks. In: IAPR TC3 Workshop on Artificial Neural Networks in Pattern Recognition. Lecture notes in artificial intelligence, Springer Verlag Berlin, Heidelberg, New York (2006)
15. Karpathy, A., Miller, S., Fei-Fei, L.: Object Discovery in 3D scenes via Shape Analysis. In: IEEE International Conference on Robotics and Automation (ICRA) (2013)
16. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view RGB-D object dataset. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1817–1824. IEEE (May 2011)
17. Marton, Z.C., Balint-Benczedi, F., Mozos, O.M., Blodow, N., Kanezaki, A., Goron, L.C., Pangercic, D., Beetz, M.: Part-Based Geometric Categorization and Object Reconstruction in Cluttered Table-Top Scenes. Journal of Intelligent & Robotic Systems (7) (Jan 2014)
18. Mian, A.S., Bennamoun, M., Owens, R.A.: Automatic correspondence for 3d modeling: An extensive review. International Journal of Shape Modeling 11(2) (2005)
19. Mueller, C.A., Pathak, K., Birk, A.: Object recognition in rgbd images of cluttered environments using graph-based categorization with unsupervised learning of shape

parts. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2248–2255. IEEE (2013)
20. Park, I.K., Germann, M., Breitenstein, M.D., Pfister, H.: Fast and automatic object pose estimation for range images on the GPU. Machine Vision and Applications pp. 1–18 (2009)
21. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA workshop on open source software. vol. 3 (2009)
22. Rabbani, T., Heuvel, F.V.D.: Efficient hough transform for automatic detection of cylinders in point clouds. In: Proceedings of the 11th Annual Conference of the Advanced School for Computing and Imaging. vol. 3, pp. 60–65 (2004)
23. Reed, R.D., Marks, R.J.: Neural smithing: supervised learning in feedforward artificial neural networks. Mit Press (1998)
24. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1–6. IEEE (2009)
25. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA). Shanghai, China (May 9-13 2011)
26. Rusu, R., Blodow, N., Marton, Z., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3384–3391. Ieee (Sep 2008)
27. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluation of color descriptors for object and scene recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8. No. 1, Ieee (Jun 2008)
28. Schaul, T., Bayer, J., Wierstra, D., Sun, Y., Felder, M., Sehnke, F., Rückstieß, T., Schmidhuber, J.: Pybrain. The Journal of Machine Learning Research 11, 743–746 (2010)
29. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in images. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. vol. 1, pp. 370–377. IEEE (2005)
30. Uckermann, A., Haschke, R., Ritter, H.: Real-time 3D segmentation of cluttered scenes for robot grasping. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012). pp. 198–203. IEEE (Nov 2012)
31. Viola, P.A., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. International Journal of Computer Vision 63(2), 153–161 (2005)
32. Wahl, E., Hillenbrand, U., Hirzinger, G.: Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification. In: Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM) (2010)
33. Woodford, O.J., Pham, M.T., Maki, A., Perbet, F., Stenger, B.: Demisting the Hough Transform for 3D Shape Recognition and Registration. International Journal of Computer Vision 106(3), 332–341 (Apr 2013)
34. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision 7(3), 119–152 (1994)